

ExplicitDP: Generalized Multi-Task Manipulation with Explicit Diffusion Policies via Phase-Aware Task Understanding

Xinmin Fang¹, Lingfeng Tao², Zhengxiong Li¹

Abstract—When a robot learns multiple manipulation tasks through a shared policy, the lack of explicit phase structure causes multi-task interference, leading to phase confusion where the robot struggles to distinguish execution phases, and goal mislocalization where the robot fails to identify correct phase-specific targets. We identify that this interference arises because, without phase-level task understanding, the policy’s latent space organizes by visual similarity, mapping visually similar states from different tasks or phases to nearby regions despite requiring conflicting actions. We observe that manipulation tasks share a compositional phase structure in operational space, where different tasks executing the same phase exhibit compatible action patterns. Automatically discovering this structure provides the robot with explicit understanding of its current phase and spatial goals, thereby resolving both failure modes. Building on this insight, we present **ExplicitDP**, a hierarchical framework that leverages end-effector motion patterns as a physics-grounded signal for phase-centric reorganization. A high-level policy organizes the continuous latent space into phase-aligned regions and generates latent commands encoding both phase identity and spatial goals, while a low-level diffusion policy conditions on this guidance to generate precise, phase-appropriate actions. We evaluate **ExplicitDP** across six challenging benchmarks, including MetaWorld MT10/MT50 and four LIBERO task suites. Results demonstrate that **ExplicitDP** achieves 64.4% average success rate, substantially outperforming multi-task baselines, including the flat multi-task diffusion policy (MT-DP, 37.8%) and the hierarchical diffusion policy (HDP, 48.1%), while approaching single-task diffusion policy performance (ST-DP, 68.4%). **ExplicitDP** also achieves 74.3% Object Manipulation Success Rate, a 20.8% gap over HDP, confirming that phase-aware task understanding effectively resolves goal mislocalization.

I. INTRODUCTION

Multi-task robotic manipulation is a central capability for general-purpose robots operating in diverse scenarios such as homes, warehouses, and laboratories [1]. Enabling one robot to accomplish a broad set of manipulation tasks with a unified policy is a long-standing goal of robot learning, where each task involves different objects, goal configurations, sequential phases, and interaction patterns. For instance, pick-and-place requires reaching, grasping, lifting, and releasing, while door-opening involves approaching, turning, and pulling. Mastering such diverse tasks with a single policy is essential for scalable deployment, yet remains a significant challenge.

*This work is partially supported by NSF Awards #2426469, #2426470, #2602731, #2436428, Colorado OEDIT AI Grant Award, and the NVIDIA Academic Grant.

¹X. Fang, and Z. Li are with the Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80217 USA (e-mail: xinmin.fang@ucdenver.edu, zhengxiong.li@ucdenver.edu)

²L. Tao is with the Department of Robotics and Mechatronics Engineering, Kennesaw State University, InDex Lab. (email: ltiao2@kennesaw.edu)

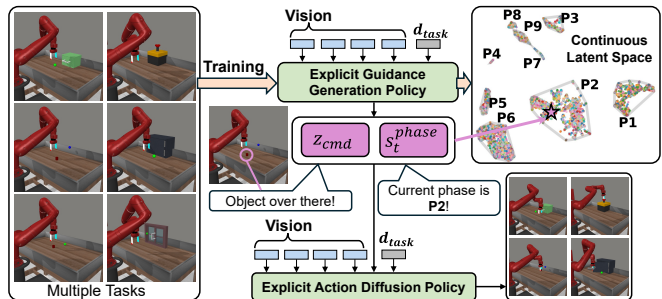


Fig. 1. ExplicitDP is a hierarchical framework that enables robots to understand the current phase and spatial goals from end-effector motion patterns, and generate precise manipulation actions across multi-task scenarios.

When a robot learns multiple manipulation tasks through a shared policy, the policy lacks understanding of the compositional phase structure underlying these tasks, conflating action patterns from different tasks and phases within a shared representation. Without phase-level task understanding, the policy exhibits two critical failure modes during execution. First, the robot exhibits *phase confusion*: it cannot reliably distinguish which execution phase it is currently in, producing phase-incompatible actions such as closing the gripper before reaching the target object. Second, the robot suffers from *goal mislocalization*: it fails to accurately identify the target position corresponding to the current task phase, causing the end-effector to deviate from the intended manipulation target [2]. These failures arise because different tasks often share visually similar observations yet demand fundamentally different actions, and a shared policy without phase-level understanding cannot reconcile these conflicting requirements. This multi-task interference is not confined to any particular learning paradigm; it represents a fundamental challenge of multi-task manipulation.

Traditional control-based approaches address multi-task manipulation through explicit task encoding with finite state machines [3] and task-and-motion planning with symbolic predicates [4]. While these methods achieve reliable performance in structured environments, they require extensive manual engineering for task decomposition and geometric modeling, hindering scalability to diverse task repertoires. Learning-based approaches reduce this modeling burden by acquiring shared policies directly from demonstrations. Deep reinforcement learning enables robots to acquire visuomotor skills through interaction, yet extending to broad multi-task settings often results in sample inefficiency, unstable training, and negative transfer [5]. Diffusion policies [6] have emerged as a promising framework for multi-task manipulation, as their capacity to model multi-modal action distributions

through iterative denoising suits the diversity of multi-task action spaces. However, this expressiveness becomes a liability in multi-task training: the learned denoising model must compromise between conflicting action distributions from different tasks, making diffusion policies susceptible to multi-task interference. Recent hierarchical approaches attempt to mitigate this: the Hierarchical Diffusion Policy (HDP) [7] and H^3DP [8] predict target states to guide low-level controllers, but their high-level models serve all tasks in a shared representation without explicit phase distinction, leaving phase confusion unresolved. SkillDiffuser [9] and Discrete Policy [10] introduce discrete skill abstractions, yet such coarse task-level skills provide limited phase-specific spatial goal information, leaving goal mislocalization undressed.

We observe that this interference arises because, without explicit phase structure, the policy’s latent space organizes primarily by visual similarity of observations. Visually similar states from different tasks or phases are mapped to nearby latent regions, yet they may require fundamentally different actions (e.g., the same tabletop scene demands reaching during approach but gripper closure during grasping), creating unresolvable conflicts in the shared action mapping. However, when viewed through the lens of end-effector motion in operational space, diverse manipulation tasks decompose into a shared vocabulary of canonical motion primitives (approaching, engaging, manipulating, releasing) dictated by workspace geometry and physical interaction constraints. Different tasks executing the same phase exhibit compatible rather than conflicting action patterns in this space. These observations motivate a principle we term *phase-aware task understanding*: automatically discovering the compositional phase structure shared across tasks provides the robot with explicit knowledge of its current phase and spatial goals, thereby resolving both failure modes. End-effector motion patterns provide a physics-grounded signal for this discovery, as phase transitions correspond directly to shifts in dominant movement axes and gripper engagement states. This insight motivates a question: *Can a robot automatically acquire phase-aware understanding of its manipulation tasks, and leverage this understanding to achieve precise multi-task execution with a single shared policy?*

We propose **ExplicitDP**, a hierarchical framework that achieves phase-aware task understanding by automatically discovering the compositional phase structure of end-effector motion. As shown in Figure 1, **ExplicitDP** achieves this through two complementary components. First, we design a structured Action Phase State (APS) representation derived from end-effector actions that captures phase-specific motion patterns across diverse tasks without requiring manual annotation of phase boundaries or waypoint specifications. Building on APS, an Explicit Guidance Generation Policy (EGGP) partitions the continuous latent space into task phase regions and generates latent commands encoding both the current phase identity and phase-specific spatial goals. Second, an Explicit Action Diffusion Policy (EADP) adopts diffusion models as the action generation backbone for their strength

in modeling multi-modal distributions, and conditions the denoising process on this explicit guidance, enabling the robot to generate phase-appropriate actions when visually similar observations demand conflicting responses across different tasks. This design separates the understanding of “what phase the robot is in and where it should go” from the generation of precise motor commands, thereby resolving both phase confusion and goal mislocalization.

We conduct comprehensive experiments across six challenging benchmarks including MetaWorld MT10/MT50 [1] and four LIBERO task suites [11]. **ExplicitDP** achieves an average success rate of 64.4%, substantially outperforming all multi-task baselines, including the flat multi-task diffusion policy (MT-DP, 37.8%), HDP (48.1%), and SkillDiffuser (23.1%), while approaching single-task diffusion policy performance (ST-DP, 68.4%). **ExplicitDP** also achieves an Object Manipulation Success Rate (OMSR) of 74.3%, a 20.8% improvement over HDP (53.5%), confirming that phase-aware guidance effectively resolves goal mislocalization. Ablation studies validate the necessity of each component, where removing any single component leads to substantial performance degradation.

Our contributions can be summarized as follows:

- We propose the principle of phase-aware task understanding: manipulation tasks share a compositional phase structure that can be automatically discovered from end-effector motion patterns in operational space, and organizing the policy representation around this structure naturally resolves both phase confusion and goal mislocalization.
- We develop a structured Action Phase State representation and a hierarchical architecture: a high-level policy partitions the continuous latent space into phase-aligned regions to generate explicit phase and goal guidance, while a low-level diffusion policy conditioned on this guidance produces phase-appropriate actions across diverse tasks.
- We demonstrate through comprehensive experiments on six challenging benchmarks that **ExplicitDP** achieves 64.4% average success rate, substantially outperforming multi-task baselines while approaching single-task performance, and attains 74.3% OMSR with a 20.8% gap over the strongest baseline.

II. RELATED WORK

Multi-Task Robotic Manipulation: Multi-task deep reinforcement learning (DRL) frameworks [5] learn policies through online interaction but suffer from negative transfer and unstable optimization across tasks. Imitation learning methods [2] learn from offline demonstrations, reducing exploration costs but often struggling with multi-modal action distributions. Diffusion policies [6] model action distributions through iterative denoising and have shown promise for manipulation. However, across these paradigms, training a shared policy on multiple tasks leads to multi-task interference [12], where the policy conflates action patterns

from different tasks and phases within a unified representation, resulting in phase confusion and goal mislocalization. ExplicitDP addresses this shared challenge by discovering the compositional phase structure of manipulation tasks from end-effector motion patterns and reorganizing the policy representation from task-centric to phase-centric, enabling precise multi-task action generation.

Task Decomposition in Robotic Manipulation: Control-based systems decompose tasks through manually designed finite-state machines, behavior trees, and task-and-motion planners [3], [4], providing explicit phase boundaries but requiring extensive engineering that hinders scalability. Learning-based approaches discover hierarchical policies or discrete skill abstractions [9], [10], [13], yet these methods primarily identify phase identity and provide limited guidance about phase-specific spatial goals. Sensor-augmented methods infer task phases from force-torque readings or robot state telemetry [14], but require specialized hardware and calibration pipelines. Recent Vision-Language-Action (VLA) models [15] segment subtasks into coarse phases, yet their binary decomposition lacks fine-grained spatial goal encoding. None of these approaches achieve the task-to-phase representational reorganization needed for phase-aware task understanding. ExplicitDP extracts phase structure directly from end-effector actions in operational space, which encode both phase identity and spatial goal information without manual annotations or additional sensors.

Hierarchical Approaches for Multi-Task Manipulation: Hierarchical target-state methods, such as HDP [7] and H³DP [8], predict target states to condition low-level controllers, but their high-level modules serve all tasks in a shared representation without explicit phase distinction, maintaining a task-centric organization that leaves phase confusion unresolved. Skill-centric methods, including SkillDiffuser [9], Discrete Policy [10], and mixture-of-experts approaches [16], introduce discrete skill or expert abstractions to indicate execution phases, yet these coarse representations lack phase-specific spatial goal information, leaving goal mislocalization unaddressed. Factorized approaches [17] decompose the policy into specialized sub-modules, but require training multiple diffusion models, hindering scalability. ExplicitDP achieves phase-centric reorganization through continuous latent commands that jointly encode current phase and spatial goals within a single unified policy, providing explicit phase-aware task understanding where existing hierarchical methods fall short.

III. METHODOLOGY

A. Hierarchical Framework Overview

ExplicitDP is a hierarchical framework that achieves phase-aware task understanding by discovering the compositional phase structure of manipulation tasks and reorganizing the shared representation from task-centric to phase-centric. As shown in Figure 2, ExplicitDP consists of an Explicit Guidance Generation Policy (EGGP, Section III-B), which leverages the Action Phase State (APS) s_t^{phase} derived from end-effector actions to reorganize the continuous latent space into

phase-centric regions and generate latent commands z^{cmd} encoding both current phase and phase-specific spatial goals, and an Explicit Action Diffusion Policy (EADP, Section III-C), which conditions on z^{cmd} to generate precise, phase-appropriate actions. Section III-D details the training strategy that leverages s_t^{phase} as privileged supervision.

B. High-Level Policy: Explicit Guidance Generation

To achieve phase-centric reorganization, the high-level policy faces two fundamental challenges. First, it needs to simultaneously encode phase information to identify the current execution stage and spatial goal information to specify target positions and gripper states, enabling the robot to select phase-appropriate actions. Second, this dual-level information should be automatically extracted from demonstrations across tasks with vastly different phase structures, without requiring manual specification of sub-goals or task-specific modules. To address these challenges, we propose Explicit Guidance Generation Policy (EGGP). EGGP comprises four components: a vision encoder that processes multi-view observations into feature representations, a language encoder that integrates task-specific information, an APS representation that captures phase-specific motion patterns from end-effector actions, and a prediction head that generates both the explicit latent command and phase prediction. Given multi-view visual observations over a history window $\mathbf{o}_{1:H} = \{\text{images}_t\}_{t=t_0-H+1}^{t_0}$ and a task description embedding $\mathbf{d}^{\text{task}}$, EGGP produces two outputs. First, it generates a latent command $z_{t_0}^{\text{cmd}} \in \mathbb{R}^{d_{\text{cmd}}}$ that encodes both current phase identity and phase-specific spatial goals, guiding the robot toward phase-appropriate manipulation. Second, it produces a phase state prediction $s_{t_0}^{\text{phase}} \in \mathbb{R}^4$ that explicitly identifies the current APS.

Action Phase State (APS): Achieving task-to-phase reorganization requires a representation that serves as the organizing signal for the latent space. Manipulation tasks exhibit diverse phase structures with varying numbers and types of phases (e.g., pick-and-place involves reaching, grasping, lifting, and releasing, while door-opening requires approaching, turning, and pulling). As established in Section I, different tasks executing the same phase exhibit compatible action patterns in operational space, where end-effector motion patterns provide a physics-grounded signal for identifying shared phases. Specifically, manipulators follow a coarse-to-fine strategy: approach motions along axes with larger workspace range, followed by fine alignment along constrained axes, concluding with gripper engagement. These motion patterns, characterized by dominant movement directions and gripper state transitions, are consistent across human demonstrations and scripted trajectories, serving as reliable, task-agnostic phase indicators.

Building on this insight, we introduce APS representation that encodes these motion patterns into a structured four-dimensional binary vector. At each timestep t , given the end-effector action $\mathbf{a}_t = (\Delta x_t, \Delta y_t, \Delta z_t, a_t^{\text{gripper}})$, we define the

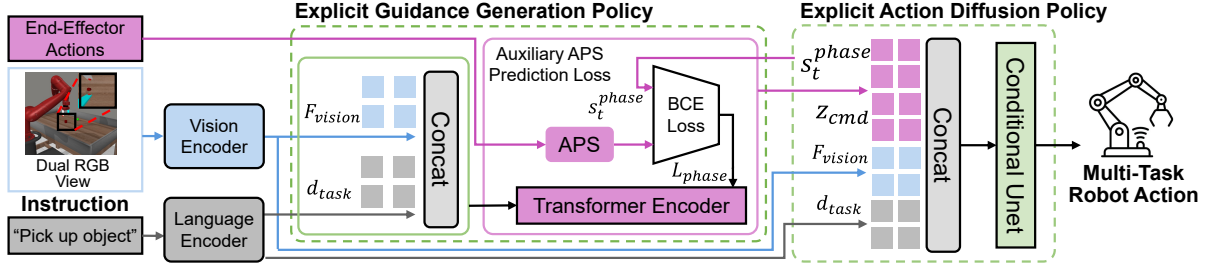


Fig. 2. ExplicitDP consists of an Explicit Guidance Generation Policy that reorganizes the latent space from task-centric to phase-centric using APS derived from end-effector motion patterns, and an Explicit Action Diffusion Policy that generates phase-appropriate actions conditioned on this guidance.

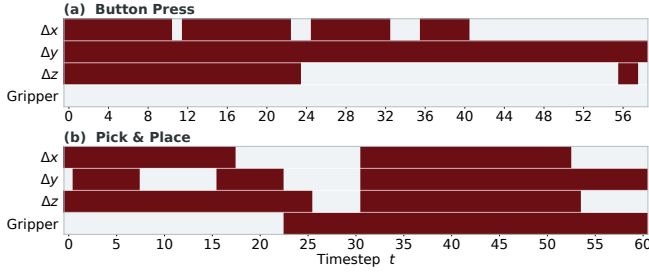


Fig. 3. Temporal APS heatmaps for task Button Press and Pick & Place. Dark regions indicate active motion along each axis or gripper engagement.

APS $s_t^{\text{phase}} \in \{0, 1\}^4$ as:

$$s_t^{\text{phase}}[i] = \begin{cases} 1, & \text{if } |\Delta p_i| > \tau_i \text{ for } i \in \{0, 1, 2\} \\ 1, & \text{if } a_t^{\text{gripper}} > \tau_{\text{grip}} \text{ for } i = 3 \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $\Delta p_i \in \{\Delta x_t, \Delta y_t, \Delta z_t\}$ for $i \in \{0, 1, 2\}$ represents motion along the x , y , and z axes respectively, and $s_t^{\text{phase}}[3]$ indicates gripper engagement. We set all thresholds to $\tau_0 = \tau_1 = \tau_2 = \tau_{\text{grip}} = 0.02$. Since actions are normalized to $[-1, 1]$, this value corresponds to 2% of the maximum per-step command and separates deliberate motion from idle jitter; the same value is used for all benchmarks and tasks without per-task tuning. This representation provides two key advantages for phase-centric reorganization. First, APS captures semantically meaningful phase structure grounded in physical motion primitives, ensuring that different tasks sharing the same phase map to compatible representations. Second, it generalizes across diverse task types without task-specific engineering, as most manipulation tasks naturally follow the coarse-to-fine motion pattern in operational space. Figure 3 visualizes the temporal APS patterns for two representative MetaWorld tasks. Button Press exhibits sustained vertical motion (Δy) with periodic lateral adjustments, whereas Pick & Place displays clearly delineated sequential phases, including approach, grasp, transport, and stop, marked by distinct end-effector state transitions. These contrasting patterns confirm that APS effectively captures task-specific phase structures from end-effector actions.

Vision Encoder: We adopt a dual-view observation setup: a global view captures the overall workspace layout for spatial reasoning, while a gripper-mounted view provides close-up observations for precise manipulation. We process these observations using ResNet-18 encoders with shared weights across both views. At each timestep t within the history window, we encode the dual view separately to produce per-

view feature vectors $\mathbf{f}_t^{\text{global}}, \mathbf{f}_t^{\text{gripper}} \in \mathbb{R}^{d_f}$, which are then concatenated to obtain the visual representation $\mathbf{f}_t$.

Language Encoder: To enable the policy to follow user instructions, we incorporate task descriptions as conditioning inputs. We encode task descriptions using the Contrastive Language-Image Pre-training (CLIP) text encoder [18], which produces a task embedding $\mathbf{d}^{\text{task}} \in \mathbb{R}^{d_{\text{clip}}}$ from natural language descriptions.

Architecture: At each timestep t within the history window, we concatenate the visual feature $\mathbf{f}_t$ with the task embedding $\mathbf{d}^{\text{task}}$ and project through a single-layer multi-layer perceptron (MLP) to obtain the transformer input: $\mathbf{e}_t = \text{MLP}_{\text{embed}}([\mathbf{f}_t, \mathbf{d}^{\text{task}}])$. We add positional encodings to each embedded token to preserve temporal ordering: $\tilde{\mathbf{e}}_t = \mathbf{e}_t + \text{PE}(t)$, where PE denotes standard sinusoidal positional encoding. The architecture processes the encoded sequence $\{\tilde{\mathbf{e}}_t\}_{t=t_0-H+1}^{t_0}$ through a multi-head Transformer encoder to capture temporal dependencies, producing contextualized representations at each timestep. We then apply average pooling along the temporal dimension to obtain a unified representation $\mathbf{h}$.

EGGP employs a projection head with dual output branches to generate explicit phase and goal guidance. The pooled representation $\mathbf{h}$ is projected through an MLP to produce a combined latent representation: $\mathbf{z} = \text{MLP}_{\text{proj}}(\mathbf{h})$, where $\mathbf{z} \in \mathbb{R}^{d_{\text{phase}} + d_{\text{cmd}}}$. This representation is then partitioned into two components: the APS prediction component $\mathbf{z}^{\text{phase}} = \mathbf{z}[0:d_{\text{phase}}] \in \mathbb{R}^{d_{\text{phase}}}$ and the command component $\mathbf{z}^{\text{cmd}} = \mathbf{z}[d_{\text{phase}}:d_{\text{phase}} + d_{\text{cmd}}] \in \mathbb{R}^{d_{\text{cmd}}}$. The APS prediction component is mapped through a linear layer to produce logits $\hat{\mathbf{s}}^{\text{phase}} \in \mathbb{R}^4$ for fundamental motion patterns, providing explicit phase information. The command component $\mathbf{z}^{\text{cmd}}$ serves as the latent command passed to the low-level policy, encoding current phase and spatial goal information.

Loss Function and Latent Space Partitioning: To achieve phase-centric reorganization of the latent space, we introduce an **auxiliary APS prediction loss** that leverages end-effector motion patterns to organize the continuous latent space into phase-aligned regions. During training, we compute the ground-truth APS s^{phase} from the demonstrated end-effector actions at the current timestep using the threshold-based criteria defined previously, and train the phase prediction head to minimize the auxiliary binary cross-entropy loss:

$$L_{\text{phase}} = \sum_{i=0}^3 \text{BCE}(s^{\text{phase}}[i], \hat{s}^{\text{phase}}[i]), \quad (2)$$

where BCE denotes binary cross-entropy since each dimension can be independently activated in the multi-hot encoding. The overall high-level policy loss is defined as $L_{hi} = \lambda_{phase} L_{phase}$, where λ_{phase} controls the strength of phase supervision.

This auxiliary loss achieves the task-to-phase reorganization of the latent space. By supervising $\mathbf{z}^{phase}$ to predict the APS, we impose an inductive bias that organizes the continuous latent space according to phase-specific motion patterns rather than task identities. During training, observations from diverse tasks exhibiting similar phases (e.g., approach phases across different manipulation tasks) are encouraged to map to nearby regions, while observations corresponding to different phases (e.g., horizontal approach versus vertical engagement) are pushed apart. This reorganization creates a structured latent space where $\mathbf{z}^{cmd}$ encodes discrete phase information through its position within the phase-organized space. Gradient signals from the low-level action prediction loss further encourage $\mathbf{z}^{cmd}$ to carry the continuous spatial goal information needed for precise, phase-appropriate action generation.

C. Low-Level Policy: Explicit Action Diffusion

Given the phase-centric latent commands from EGGP, the remaining challenge is translating this phase understanding into precise actions. We propose the Explicit Action Diffusion Policy (EADP), which conditions a diffusion process on the guidance to generate phase-appropriate robot actions.

Architecture: EADP comprises two components: a conditioning module that fuses visual features, latent commands, and APS; a conditional U-Net that processes the noisy action sequence. EADP takes as input visual features $\{\mathbf{f}_t\}$, the latent command $\mathbf{z}^{cmd}$, and the APS $\mathbf{s}_t^{phase}$, and produces the action sequence $\mathbf{A} = \{\mathbf{a}_{t_0+1}, \dots, \mathbf{a}_{t_0+H_a}\}$ for robot execution. The conditioning module concatenates these inputs to form a conditioning vector $\mathbf{g}_t = [\mathbf{f}_t, \mathbf{z}_t^{cmd}, \mathbf{s}_t^{phase}]$. This unified conditioning vector is processed by a conditional 1D U-Net architecture with encoder dimensions [256, 512] that operates along the temporal dimension of the action sequence. The U-Net processes the noisy action sequence $\tilde{\mathbf{A}} \in \mathbb{R}^{H_a \times d_a}$ through a series of encoder and decoder blocks, where H_a denotes the action horizon and d_a represents the action dimension. To inject explicit guidance into the denoising process, we employ Feature-wise Linear Modulation (FiLM) [19] within each U-Net block, which modulates the intermediate feature representations based on the global conditioning vector $\mathbf{g}_t$; adapting the internal representations to the explicit phase and spatial goal information enables EADP to resolve both phase confusion and goal mislocalization.

The final U-Net output produces the predicted noise $\hat{\epsilon} \in \mathbb{R}^{H_a \times d_a}$, which parameterizes the reverse diffusion process. Through iterative denoising steps, EADP generates the final action sequence $\mathbf{A}$ that specifies robot end-effector commands, which is supervised by ground-truth demonstrated actions during training.

D. Training and Deployment

ExplicitDP employs a training strategy that leverages APS as privileged supervision to achieve phase-centric latent space

TABLE I
KEY HYPERPARAMETERS FOR EXPLICITDP IMPLEMENTATION.

Hyperparameter	Value
Observation history window H	2
Action horizon H_a	8
Visual feature dimension d_f	512
Latent dimension d_z	512
APS prediction dimension d_{phase}	64
Command dimension d_{cmd}	448
Transformer layers	8
Transformer attention heads	8
Diffusion denoising steps	100
Phase loss weight λ_{phase}	0.2
Learning rate	10^{-4}

organization. Table I summarizes the key hyperparameters. The total training objective combines high-level and low-level losses: $L_{total} = L_{lo} + \lambda_{hi} L_{hi}$, where $L_{lo} = \mathbb{E}[\|\hat{\epsilon} - \epsilon\|_2^2]$ denotes the standard Denoising Diffusion Probabilistic Model (DDPM) [20] noise prediction loss and λ_{hi} balances the high-level phase supervision contribution. During deployment, the system operates solely from visual observations and task descriptions, without access to end-effector actions or phase annotations. EGGP predicts the latent command $\mathbf{z}_t^{cmd}$ containing phase and goal information along with the current $\hat{\mathbf{s}}_t^{phase}$ from visual inputs, while EADP conditions on this guidance to generate actions. This training-deployment asymmetry enables ExplicitDP to learn phase-centric latent representations from privileged end-effector information during training while maintaining fully autonomous multi-task manipulation at deployment.

IV. EXPERIMENTAL SETUP

Benchmark: We evaluate ExplicitDP on multiple challenging benchmarks: MetaWorld MT10 and MT50 [1], and four LIBERO task suites [11]. MetaWorld is widely adopted in robotic manipulation research; MT10 comprises 10 diverse tasks, while MT50 extends to 50 tasks for a simulated Sawyer arm. Both require a single policy to master all tasks, posing significant multi-task challenges. The four LIBERO suites (LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long) complement MetaWorld by providing diverse task structures and object types.

Dataset: For training and evaluating ExplicitDP, our datasets consist of expert demonstrations collected for all tasks within the evaluated benchmarks. We collected 50 demonstration episodes for each MetaWorld task using the native expert Sawyer policy provided by MetaWorld [1]. For LIBERO suites, we use the provided 50 expert demonstrations for each task. Each demonstration includes dual RGB image sequences and the corresponding robot actions.

Baselines: We benchmark ExplicitDP against the following baselines. First, a **Single-Task Diffusion Policy (ST-DP)** consists of standard transformer-based diffusion policies [6] with our vision encoder, each trained independently for one task, establishing a strong performance reference for specialized models. Second, a **Multi-Task Diffusion Policy (MT-DP)** extends ST-DP to the multi-task setting through the

TABLE II
SUCCESS RATES (%) OF ALL METHODS ACROSS ALL BENCHMARKS AND THREE SEEDS.

Method	MT10	MT50	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Avg
ST-DP (Single Task)	75.5	68.3	71.2	69.8	66.4	58.9	68.4 $\pm$ 5.1
MT-DP	48.9	31.7	42.3	39.7	35.9	28.2	37.8 $\pm$ 6.8
HDP [7]	60.0	43.2	51.3	50.8	45.1	38.1	48.1 $\pm$ 7.0
SkillDiffuser [9]	23.2	19.4	28.1	25.4	22.8	19.6	23.1 $\pm$ 3.1
ExplicitDP (Ours)	73.5	61.8	67.9	66.3	62.1	54.7	64.4$\pm$5.8

language encoder. This baseline directly reveals the multi-task interference that arises when a shared policy learns multiple tasks without phase-aware task understanding. We also include two recent hierarchical diffusion policies that aim to mitigate multi-task interference through structured guidance. **SkillDiffuser** by Liang et al. [9] represents approaches that learn discrete and interpretable skills as subgoals. **Hierarchical Diffusion Policy (HDP)** by Ma et al. [7] uses explicit geometric poses as subgoals, allowing comparison with a hierarchical method that relies on concrete geometric guidance rather than phase-centric latent guidance. Since HDP has not fully open-sourced all of its training code, we reproduce it based on the authors’ instructions [21] and validate our implementation by matching the MT10 performance reported in the original paper. Discrete Policy [10] is excluded because it is not open-sourced, and transformer-based approaches such as RT-1/RT-2 require large-scale pre-training beyond our experimental scope.

Metrics: We evaluate ExplicitDP using two complementary metrics. First, **Task Success Rate** measures the percentage of successful task completions based on each benchmark’s predefined success criteria. Second, **Object Manipulation Success Rate (OMSR)** measures the percentage of episodes in which the robot correctly localizes and manipulates the target object, regardless of whether the full task subsequently succeeds. Object manipulation is judged by each benchmark’s native evaluation signals: MetaWorld exposes per-step indicators for reaching and grasping the target object (*near_object* and *grasp_success*), and LIBERO provides object-centric goal predicates that verify contact with and displacement of the target object. An episode counts toward OMSR when these object-level criteria are satisfied, so OMSR isolates goal mislocalization from failures in later execution.

V. EVALUATION AND EXPERIMENTS

A. Comparison with Baselines

We evaluate ExplicitDP across six benchmarks spanning MetaWorld MT10/MT50 and four LIBERO suites, encompassing diverse manipulation primitives and complexity levels to assess phase-aware task understanding. As shown in Table II, ExplicitDP achieves an overall average of 64.4%, substantially outperforming all multi-task baselines while approaching single-task performance (68.4% for ST-DP). On MT50, ExplicitDP achieves 61.8% success rate, representing an 18.6% gap over HDP (43.2%). This margin demonstrates that ExplicitDP effectively discovers compositional phase structure across tasks, enabling scalability to larger task sets. Compared to existing multi-task methods, ExplicitDP shows

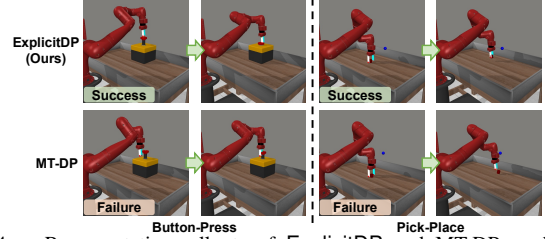


Fig. 4. Representative rollouts of ExplicitDP and MT-DP on button-press-topdown and pick-place: ExplicitDP reaches and manipulates the target object, whereas MT-DP suffers goal mislocalization, with the gripper deviating from the target.

substantial improvements: 42.4% over SkillDiffuser (19.4%) and 16.1% over HDP (48.1%). We note that SkillDiffuser underperforms MT-DP, which we attribute to its discrete skills providing limited phase information while leaving goal mislocalization unresolved. We validated our SkillDiffuser reproduction against the original paper and code, achieving comparable MT10 performance to their reported results.

Across LIBERO benchmarks, ExplicitDP consistently outperforms all baselines. On LIBERO-Spatial (67.9%), LIBERO-Object (66.3%), LIBERO-Goal (62.1%), and LIBERO-Long (54.7%), our method achieves an average gap of 21.7% over MT-DP. Moreover, ExplicitDP substantially surpasses SkillDiffuser by 38.8% and HDP by 10.3% on average across LIBERO suites, demonstrating the effectiveness of phase-aware task understanding across diverse manipulation scenarios.

Notably, while multi-task baselines exhibit substantial performance gaps relative to single-task policies (20.3% for HDP, 45.3% for SkillDiffuser), ExplicitDP maintains competitive performance with only a 4.0% average gap. This remaining gap should be weighed against the cost of task specialization: ST-DP maintains one policy per task, amounting to 50 separate models on MT50 alone, with training and storage costs growing linearly in the number of tasks and no knowledge shared among them; every new task must be learned from scratch. ExplicitDP serves all tasks with a single set of weights and, as shown in Section V-E, requires fewer demonstrations per task to reach strong performance.

B. Resolving Goal Mislocalization

To validate that ExplicitDP effectively resolves goal mislocalization, we measure the OMSR during task execution on MT50. Figure 4 illustrates representative cases from *button-press-topdown* and *pick-place* tasks, contrasting successful object manipulation with goal mislocalization failures. In goal mislocalization cases, the robot’s gripper deviates from the target object location, leading to manipulation failure despite appropriate visual observations. This failure mode

TABLE III

SUCCESS RATES (%) OF ABLATION STUDY ACROSS ALL BENCHMARKS AND THREE SEEDS.

Ablation	MT10	MT50	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Avg
w/o EGGP	48.5	32.8	40.7	38.9	36.2	29.8	37.8±6.0
w/o EADP	41.1	28.7	36.4	34.2	32.1	27.8	33.4±4.5
w/o APS	59.6	42.1	49.2	47.8	45.5	40.1	47.4±6.3
w/o $\mathbf{z}^{\text{cmd}}$	52.8	36.2	43.1	41.5	39.8	33.2	41.1±6.2
w/o $\mathbf{d}^{\text{task}}$	71.2	58.9	65.7	64.1	60.1	52.3	62.1±5.9
Raw End-Effector Actions	65.9	52.8	59.7	58.1	54.8	46.6	56.3±6.0
ExplicitDP (Full)	73.5	61.8	67.9	66.3	62.1	54.7	64.4±5.8

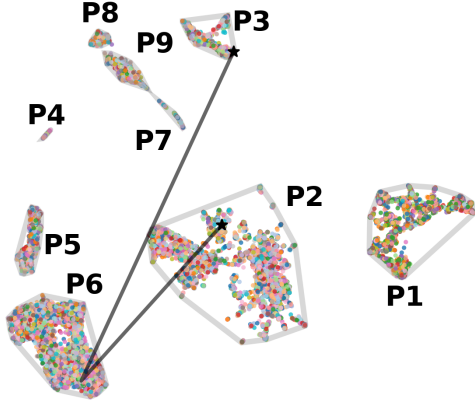


Fig. 5. UMAP projection of $\mathbf{z}^{\text{cmd}}$ on MT50. The points are latent commands, colored by task (20-hue cycle). Hierarchical density-based clustering (HDBSCAN) segments the space into nine semantic manifolds **P1-P9**, matching the APS configurations observed in demonstrations. A representative trajectory is overlaid as a black arrow with star-shaped endpoints.

occurs when multi-task interference causes the shared policy to compromise between conflicting action distributions, preventing the robot from identifying the correct target position for the current phase. Statistical analysis reveals that ExplicitDP achieves an OMSR of 74.3%, a 20.8% gap over HDP (53.5%). Compared to MT-DP (42.1%) and SkillDiffuser (28.6%), ExplicitDP demonstrates superior object localization accuracy with gaps of 32.2% and 45.7% respectively, confirming that explicit phase and goal guidance substantially improves spatial accuracy.

C. Explicit Phase Decomposition Visualization

To analyze how EGGP reorganizes the latent space from task-centric to phase-centric, we collect 20,000 $\mathbf{z}^{\text{cmd}}$ samples from MT50 validation roll-outs. Figure 5 reveals nine compact manifolds **P1-P9** in the Uniform Manifold Approximation and Projection (UMAP) embeddings, demonstrating that EGGP automatically organizes the continuous latent space into distinct phase regions without requiring manual specification of phase boundaries. Although the four-dimensional binary APS admits 16 possible configurations, only a subset arises in the demonstrations; combinations such as simultaneous large motion along all three axes during gripper closure rarely occur in practice. The nine manifolds correspond to the phase configurations that demonstrated trajectories actually visit. Critically, the heavy interleaving of task colors within each manifold confirms that $\mathbf{z}^{\text{cmd}}$ organizes by phase semantics shared across tasks rather than task identity, providing evidence of the task-to-phase reorganization. A representative trajectory from the *push* task (black arrow) transitions across **P2** to **P6** to **P3**, correspond-

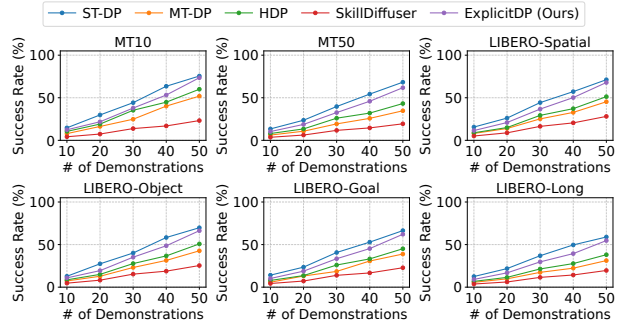


Fig. 6. Performance comparison on different numbers of demonstrations across MT10, MT50, and LIBERO suites.

ing to approach, align, and push phases, confirming that $\mathbf{z}^{\text{cmd}}$ encodes coherent phase progression. This phase-centric organization enables latent commands to encode phase information alongside spatial goal information; consistent with this interpretation, removing $\mathbf{z}^{\text{cmd}}$ lowers the average success rate from 64.4% to 41.1% (Table III).

D. Ablation Studies

We conduct ablation studies to validate the necessity of each component, with results across all benchmarks presented in Table III. Removing the EGGP (reducing to a simple MLP) degrades performance substantially, with average success rates dropping from 64.4% to 37.8%, demonstrating that EGGP is critical for phase-centric reorganization. Replacing EADP with a simple MLP also results in severe performance degradation (33.4%), confirming that conditioning the action generation on phase-centric guidance is essential for translating the robot’s phase understanding into precise motor control. Besides, disabling the APS mechanism reduces average performance to 47.4%, highlighting its role as the organizing signal for phase-centric latent space reorganization.

We further evaluate three conditioning variants: (1) **w/o $\mathbf{z}^{\text{cmd}}$** removes the latent command while retaining only APS for conditioning, yielding 41.1% average success. (2) **w/o $\mathbf{d}^{\text{task}}$** replaces language task descriptions with discrete one-hot task identifiers, reaching 62.1%. (3) **Raw End-Effector Actions** uses raw end-effector action sequences as phase supervision instead of our APS representation, achieving 56.3%. Removing $\mathbf{z}^{\text{cmd}}$ causes severe degradation, demonstrating that continuous latent commands are essential for explicit action guidance. Discrete task IDs perform slightly below language conditioning, suggesting that language offers improved generalization. The raw action variant (56.3%) underperforms APS, confirming that structured phase abstraction provides more effective supervision than raw action sequences.

E. Sample Efficiency Evaluation

We evaluate sample efficiency by varying demonstrations from 10 to 50 per task (Figure 6). With only 20-30 demonstrations, ExplicitDP often reaches 80% of its final performance on MT10 and achieves substantial performance on MT50, while other multi-task methods struggle even with full datasets. On LIBERO-Long tasks, ExplicitDP reaches meaningful performance ($> 40\%$) with just 20 demonstrations, while baseline methods remain markedly lower even with 50 demonstrations. This sample efficiency stems from phase-centric reorganization: tasks sharing the same phase are aligned in the latent space, so knowledge transfers rapidly across related phases.

VI. DISCUSSION AND KEY TAKEAWAYS

Phase-centric organization stabilizes training across conflicting tasks. Semantically paired tasks (e.g., *window-open* and *window-close*) exhibit oscillatory performance under task-centric policies, as the shared representation alternately specializes toward conflicting action distributions. Organizing the latent space by phase resolves this instability: complementary tasks share similar phase structures despite opposing actions, so the policy maintains distinct action patterns within the same phase region and converges stably.

Physical priors offer an inductive bias complementary to model scaling. While a natural approach to conflicting multi-task distributions is to scale up model capacity, ExplicitDP pursues an orthogonal direction: leveraging physical priors embedded in end-effector motion patterns to explicitly decompose the problem into phase understanding and action generation. This enables a compact model to approach single-task performance, suggesting that physics-grounded inductive biases and model scaling are complementary.

VII. CONCLUSION AND FUTURE WORK

We reveal that multi-task interference in robotic manipulation stems from the policy’s lack of phase-level task understanding, and propose ExplicitDP, a hierarchical framework that discovers the compositional phase structure of manipulation tasks and reorganizes the shared representation from task-centric to phase-centric. ExplicitDP leverages end-effector motion patterns as a physics-grounded signal for this discovery through a structured Action Phase State representation. With explicit understanding of its current phase and spatial goals, the robot simultaneously avoids phase confusion and goal mislocalization, generating precise, phase-appropriate actions across diverse tasks. Comprehensive experiments demonstrate a 64.4% average success rate, substantially outperforming multi-task baselines while approaching single-task performance, and a 74.3% object manipulation success rate, a 20.8% gap over the strongest baseline. Future work could extend the Action Phase State to richer motion primitives for dexterous manipulation, explore cross-embodiment transfer through shared phase-centric latent spaces, and validate the framework on real-world robotic systems.

REFERENCES

- [1] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, “Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning,” in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.
- [2] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, “Implicit behavioral cloning,” in *Conference on robot learning*. PMLR, 2022, pp. 158–168.
- [3] M. Colledanchise and L. Natale, “On the implementation of behavior trees in robotics,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5929–5936, 2021.
- [4] L. P. Kaelbling and T. Lozano-Pérez, “Integrated task and motion planning in belief space,” *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, 2013.
- [5] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman, “Scaling up multi-task robotic reinforcement learning,” in *Conference on Robot Learning*, vol. 164. PMLR, 2022, pp. 557–575.
- [6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2024.
- [7] X. Ma, S. Patidar, I. Houghton, and S. James, “Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation,” in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024, pp. 18 081–18 090.
- [8] Y. Lu, Y. Tian, Z. Yuan, X. Wang, P. Hua, Z. Xue, and H. Xu, “H³dp: Triply-hierarchical diffusion policy for visuomotor learning,” *arXiv preprint arXiv:2505.07819*, 2025.
- [9] Z. Liang, Y. Mu, H. Ma, M. Tomizuka, M. Ding, and P. Luo, “Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution,” in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024, pp. 16 467–16 476.
- [10] K. Wu, Y. Zhu, J. Li, J. Wen, N. Liu, Z. Xu, and J. Tang, “Discrete policy: Learning disentangled action space for multi-task robotic manipulation,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8811–8818.
- [11] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, “Libero: Benchmarking knowledge transfer for lifelong robot learning,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.
- [12] Z. Dong, Y. Liu, Y. Li, H. Zhao, and J. Hao, “Conditioning matters: Training diffusion policies is faster than you think,” in *Advances in Neural Information Processing Systems*, vol. 38, 2025.
- [13] H. Le, N. Jiang, A. Agarwal, M. Dudík, Y. Yue, and H. Daumé III, “Hierarchical imitation and reinforcement learning,” in *International conference on machine learning*. PMLR, 2018, pp. 2917–2926.
- [14] K. Chen, S. Xie, Z. Ma, P. R. Sanketi, and K. Goldberg, “Robo2vlm: Visual question answering from large-scale in-the-wild robot manipulation datasets,” *arXiv preprint arXiv:2505.15517*, 2025.
- [15] Y. Fan, S. Bai, X. Tong, P. Ding, Y. Zhu, H. Lu, F. Dai, W. Zhao, Y. Liu, S. Huang, Z. Fan, B. Chen, and D. Wang, “Long-VLA: Unleashing long-horizon capability of vision language action model for robot manipulation,” in *Proceedings of The 9th Conference on Robot Learning*, vol. 305. PMLR, 2025, pp. 2018–2037.
- [16] C. Hao, X. Zhai, Y. Liu, and H. Soh, “Abstracting robot manipulation skills via mixture-of-experts diffusion policies,” *arXiv preprint arXiv:2601.21251*, 2026.
- [17] C. Liu, H. Chen, S. H. Høeg, S. Yao, Y. Li, K. Hauser, and Y. Du, “Flexible multitask learning with factorized diffusion policy,” *arXiv preprint arXiv:2512.21898*, 2025.
- [18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [19] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [20] J. Ho, A. Jain, and P. Abbeel, “Denosing diffusion probabilistic models,” *arXiv preprint arXiv:2006.11239*, 2020.
- [21] Dyson-AI, “HDP official implementation,” 2024. [Online]. Available: <https://github.com/dyson-ai/hdp>